Advanced Ad Hoc Reporting

DL3400-1, DL3400-2

Jason Miller

Technical Consultant
Infinite Campus

Advanced Ad Hoc Reporting

DL3400-3

Marc Baker

Training Specialist
Infinite Campus

Agenda

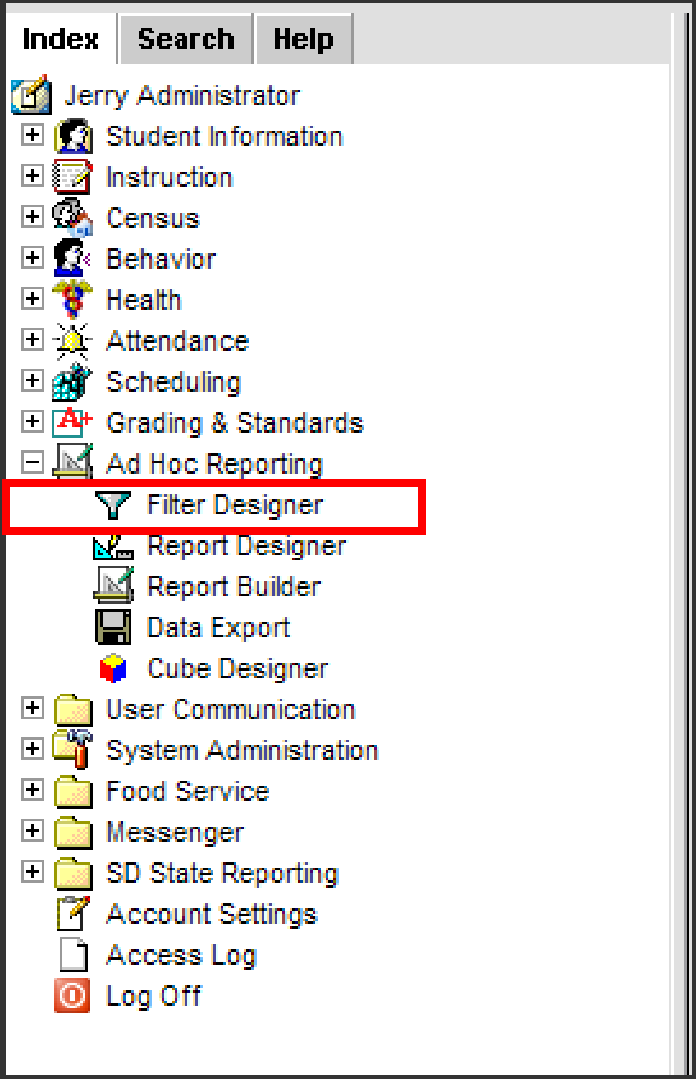
- Getting started
  - Review of Campus Ad Hoc Reporting Filter Designer
- The basic grammar of SQL
- Creating pass-through SQL queries

Getting Started

- Pass-through Structured Query Language (SQL) query filters can be created on any data type
- To create this filter
  - Have experience with Campus Ad Hoc Filter Designer
    - Selection Editor
    - Query Wizard
  - Reference the Campus schema
  - Understand SQL

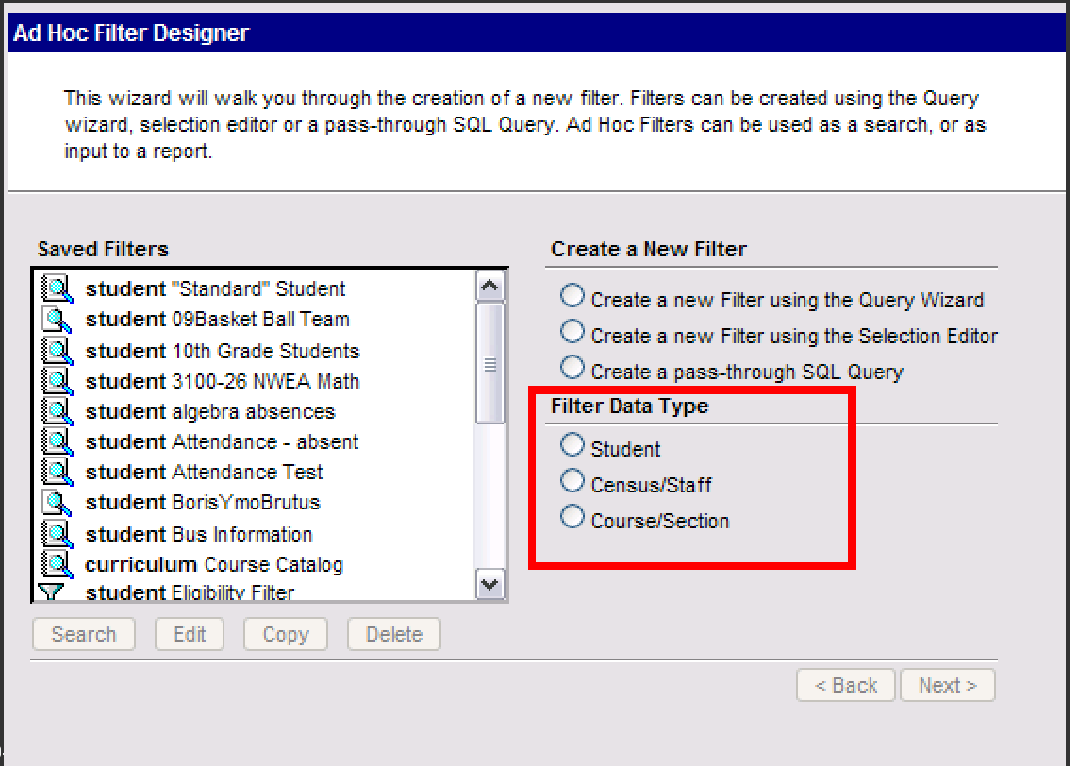
Navigation

Ad Hoc Reporting
Filter Designer



Campus Ad Hoc Filter Designer

- Three data types
  - Student
  - Census/Staff (person)
  - Course/Section (curriculum)



Ad Hoc Filter Designer

This wizard will walk you through the creation of a new filter. Filters can be created using the Query wizard, selection editor or a pass-through SQL Query. Ad Hoc Filters can be used as a search, or as input to a report.

Saved Filters

- **student** "Standard" Student
- **student** 09Basket Ball Team
- **student** 10th Grade Students
- **student** 3100-26 NWEA Math
- **student** algebra absences
- **student** Attendance - absent
- **student** Attendance Test
- **student** BorisYmoBrutus
- **student** Bus Information
- **curriculum** Course Catalog
- **student** Eligibility Filter

Create a New Filter

○ Create a new Filter using the Query Wizard
○ Create a new Filter using the Selection Editor
○ Create a pass-through SQL Query

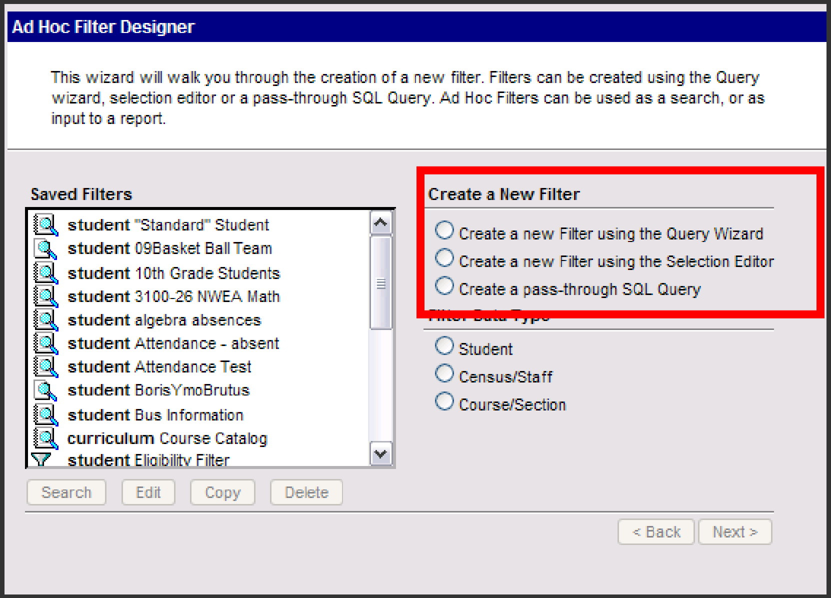
Filter Data Type

○ Student
○ Census/Staff
○ Course/Section

DL3400-1, DL3400-2. DL3400

Campus Ad Hoc Filter Designer

- Three filtering methods
  - Query Wizard
  - Selection Editor
  - Pass-through SQL Query
- All filters can be saved and reused
- Filters can be made by you and shared



Ad Hoc Filter Designer

This wizard will walk you through the creation of a new filter. Filters can be created using the Query wizard, selection editor or a pass-through SQL Query. Ad Hoc Filters can be used as a search, or as input to a report.

Saved Filters

- **student** "Standard" Student
- **student** 09Basket Ball Team
- **student** 10th Grade Students
- **student** 3100-26 NWEA Math
- **student** algebra absences
- **student** Attendance - absent
- **student** Attendance Test
- **student** BorisYmoBrutus
- **student** Bus Information
- **curriculum** Course Catalog
- **student** Eligibility Filter

Create a New Filter

○ Create a new Filter using the Query Wizard
○ Create a new Filter using the Selection Editor
○ Create a pass-through SQL Query

Filter Data Type

○ Student
○ Census/Staff
○ Course/Section

DL3400-1, DL3400-2. DL3400-3

Filtering Methods

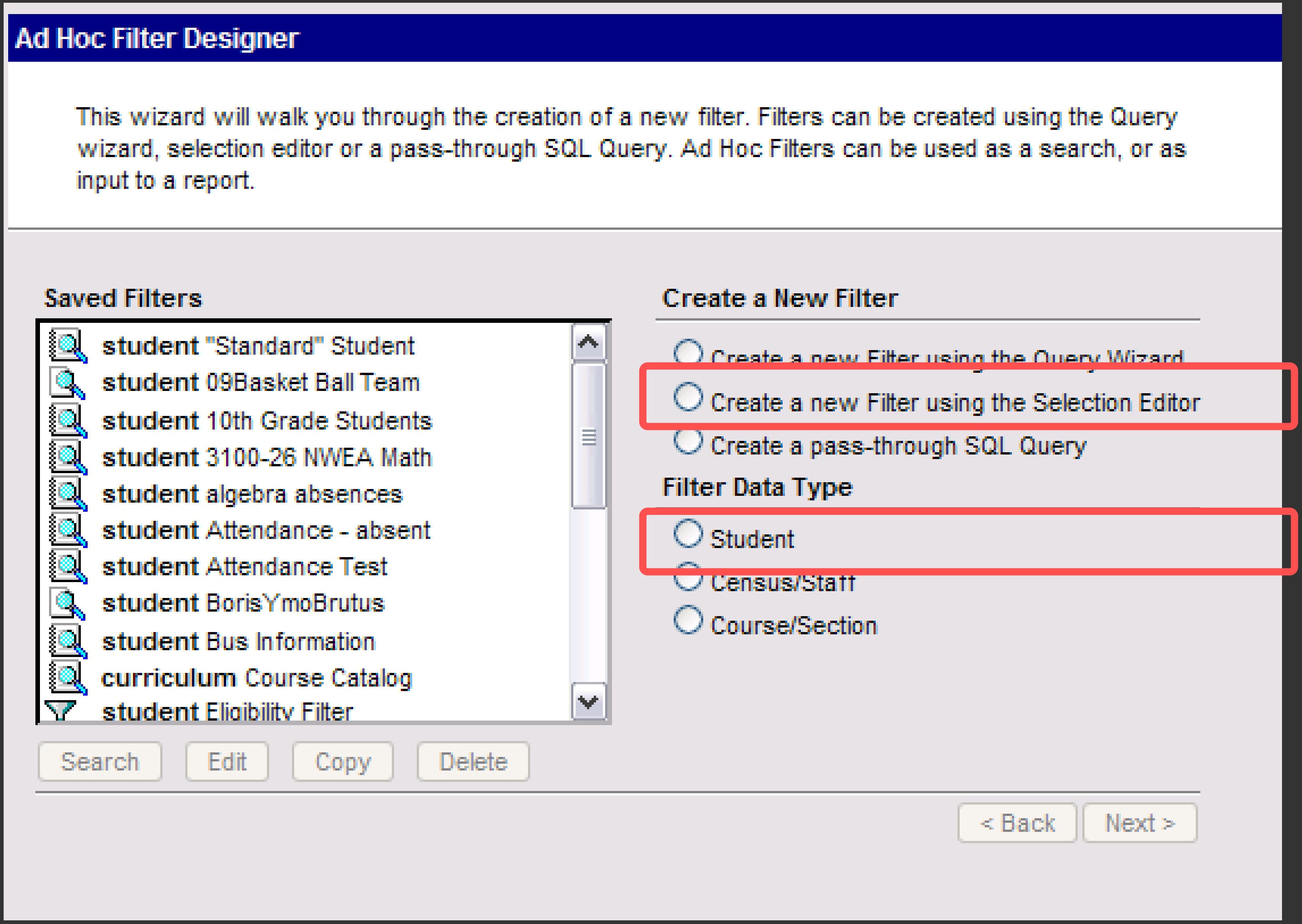
- Query Wizard
  - Creates a dynamic filter, updates automatically
  - Uses Campus field names, not plain language
  - Requires knowledge of search operators
    - Examples
      - =, <>, etc
      - NULL
      - SOUNDS LIKE
- Selection Editor
  - Only works on student data
  - Creates a static filter, must be updated manually
- Pass-through SQL Query
  - Most powerful and flexible
  - Best way to build cohort-based, longitudinal filters
  - Requires knowledge of SQL programming language

DL3400-1, DL3400-2. DL3400-3

Using Filter Designer: Selection Editor

- What is Selection Editor good for?
  - Athletic rosters
  - Activity rosters
  - Your ideas?
- Remember
  - Selection Editor filters are static
    - If a student adds or drops from the activity, the filter must be updated manually

Using Filter Designer: Selection Editor



Ad Hoc Filter Designer

This wizard will walk you through the creation of a new filter. Filters can be created using the Query wizard, selection editor or a pass-through SQL Query. Ad Hoc Filters can be used as a search, or as input to a report.

Saved Filters

- student "Standard" Student
- student 09Basket Ball Team
- student 10th Grade Students
- student 3100-26 NWEA Math
- student algebra absences
- student Attendance - absent
- student Attendance Test
- student BorisYmoBrutus
- student Bus Information
- curriculum Course Catalog
- student Eligibility Filter

Create a New Filter

○ Create a new Filter using the Query Wizard

○ Create a new Filter using the Selection Editor

○ Create a pass-through SQL Query

Filter Data Type

○ Student

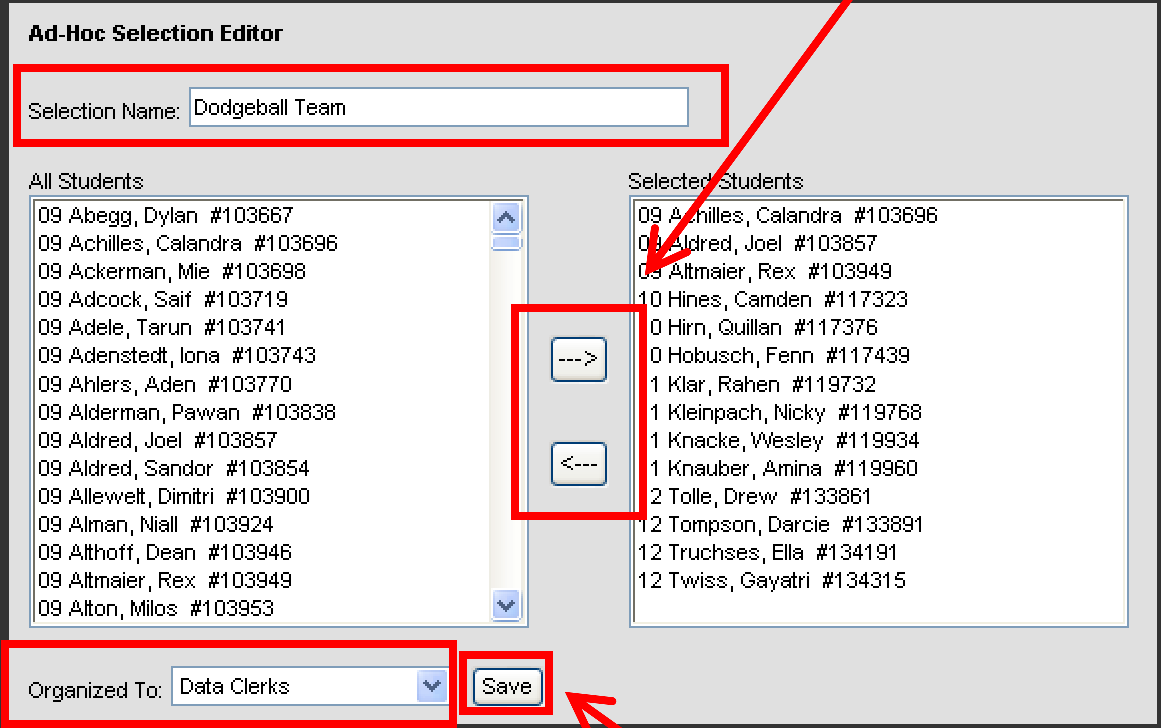
○ Census/Staff

○ Course/Section

Using Filter Designer: Selection Editor

1 Give selection a meaningful name

2 Click on a student in All Students, use right arrow to move them to Selected Students. Use left arrow to remove them from Selected Students



Ad-Hoc Selection Editor

Selection Name: Dodgeball Team

All Students

09 Abegg, Dylan #103667
09 Achilles, Calandra #103696
09 Ackerman, Mie #103698
09 Adcock, Saif #103719
09 Adele, Tarun #103741
09 Adenstedt, Iona #103743
09 Ahlers, Aden #103770
09 Alderman, Pawan #103838
09 Aldred, Joel #103857
09 Aldred, Sandor #103854
09 Allewelt, Dimitri #103900
09 Alman, Niall #103924
09 Althoff, Dean #103946
09 Altmaier, Rex #103949
09 Alton, Milos #103953

Selected Students

09 Achilles, Calandra #103696
09 Aldred, Joel #103857
09 Altmaier, Rex #103949
10 Hines, Camden #117323
10 Hirn, Quillan #117376
10 Hobusch, Fenn #117439
11 Klar, Rahen #119732
11 Kleinpach, Nicky #119768
11 Knacke, Wesley #119934
11 Knauber, Amina #119960
12 Tolle, Drew #133861
12 Tompson, Darcie #133891
12 Truchses, Ella #134191
12 Twiss, Gayatri #134315

Organized To: Data Clerks

3 Choose to save to your user account or your user group

4 Save

DL3400-1, DL3400-2. DL3400-3

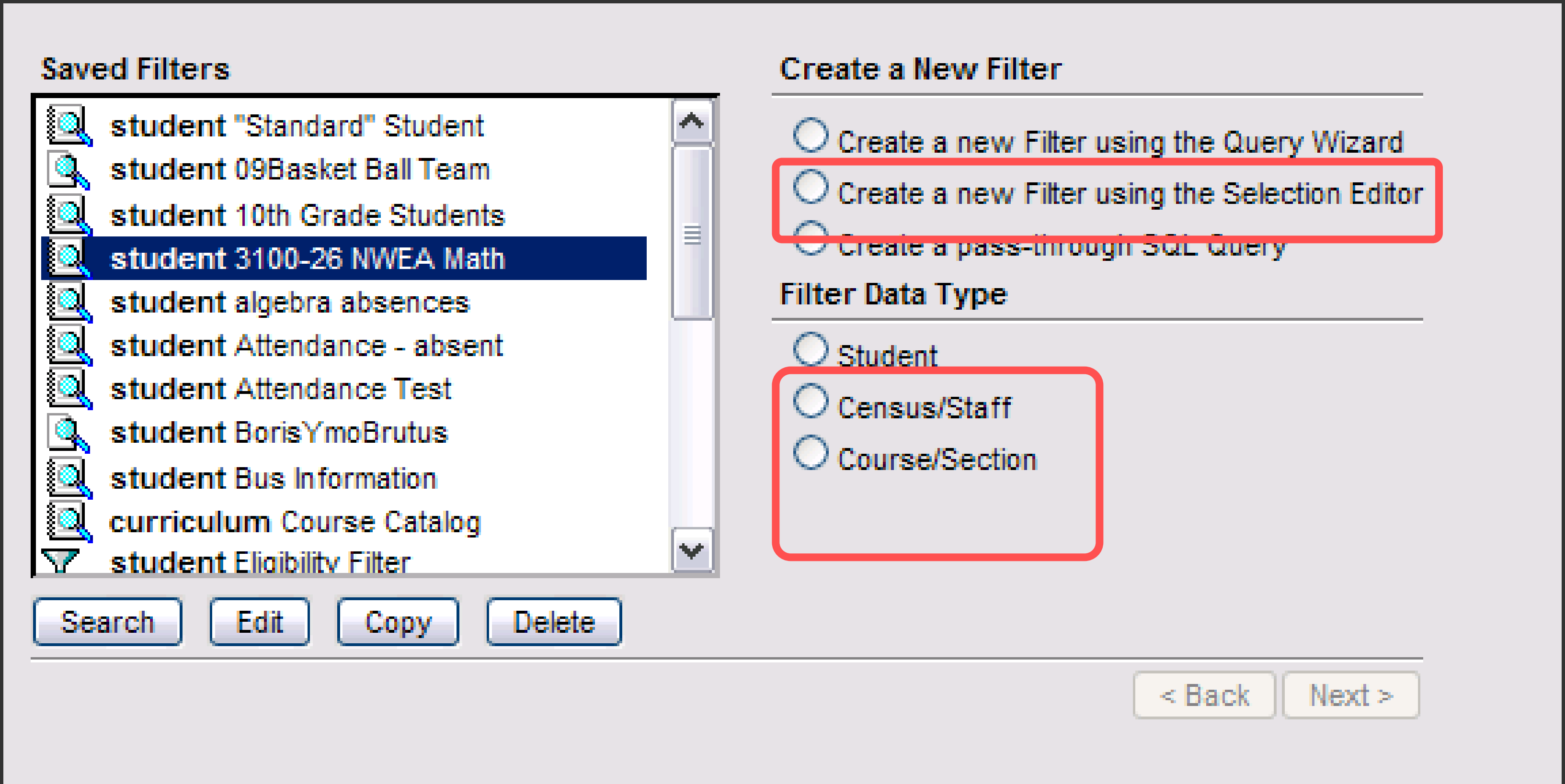
Try It: Selection Editor

- Create a Selection Editor filter
  - Filter name = SDXX##
    - XX = your initials
    - ## = last two digits of your phone number
  - Select 10 students
  - Save filter to your user account
- Check your work
  - Reload by clicking on Filter Designer
  - Check to see if your filter is in the Saved Filter list
  - Select it and click Edit to view it
  - Reload Filter Designer when done

Using Filter Designer: Query Wizard

- Query Wizard allows the creation of dynamic custom searches on any of the three data types
- What is Query Wizard good at finding?
  - Persons with information that shouldn’t be there
  - Persons who are missing information
    - Ex: Parent/guardians without phone numbers
  - Persons who meet search criteria
    - Ex: Students with course grades of D or F
    - Ex.: Students with GPA above 3.5
  - Your ideas?

Filter Designer: Query Wizard



Saved Filters

- student "Standard" Student
- student 09Basket Ball Team
- student 10th Grade Students
- student 3100-26 NWEA Math
- student algebra absences
- student Attendance - absent
- student Attendance Test
- student BorisYmoBrutus
- student Bus Information
- curriculum Course Catalog
- student Eligibility Filter

Create a New Filter

○ Create a new Filter using the Query Wizard

○ Create a new Filter using the Selection Editor

○ Create a pass-through SQL Query

Filter Data Type

○ Student

○ Census/Staff

○ Course/Section

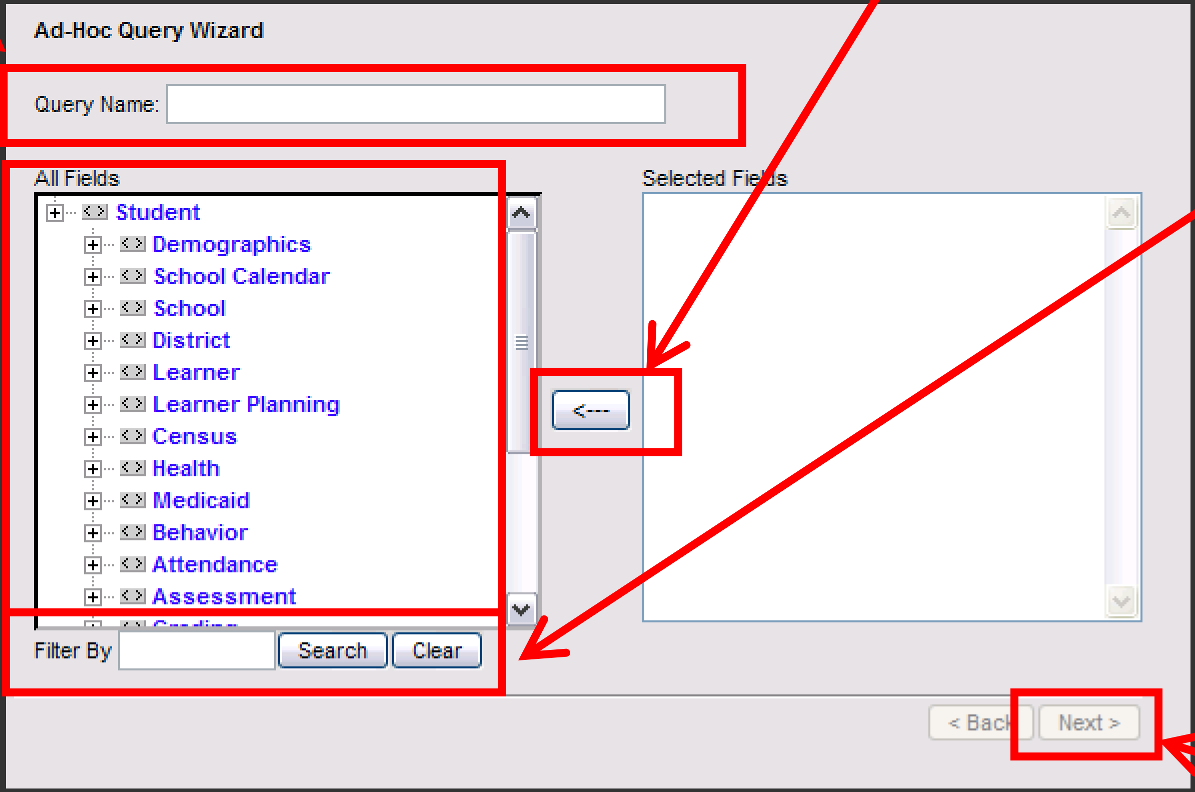
DL3400-1, DL3400-2. DL3400-3

Using Filter Designer: Query Wizard

1 Give query a meaningful name

2 Click any field that will be used as filter criteria or will be displayed on the output

To remove a field from Selected Fields, select it and click the arrow button



To filter All Fields list, enter criteria and click Search

To return to All Fields list, click Clear

Fields in the All Fields list expand, making thousands of possible filter combinations

3 Next

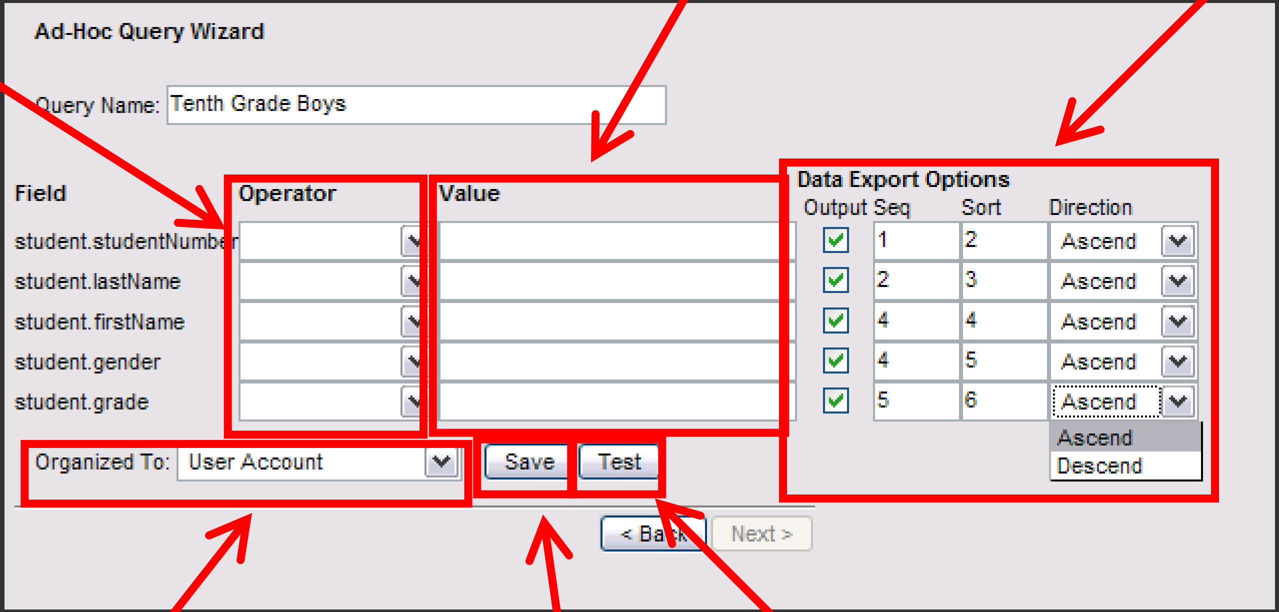
DL3400-1, DL3400-2. DL3400-3

Using Filter Designer: Query Wizard

1 Use operators to narrow your search

2 Enter filter criteria here

For export: Set options



Ad-Hoc Query Wizard

Query Name: Tenth Grade Boys

Field	Operator	Value	Data Export Options Output	Seq	Sort	Direction
student.studentNumber			☑	1	2	Ascend
student.lastName			☑	2	3	Ascend
student.firstName			☑	4	4	Ascend
student.gender			☑	4	5	Ascend
student.grade			☑	5	6	Ascend

Organized To: User Account

3 Use Organized To droplist to save the filter to your user account or share it with other user groups you belong to

4 Save

5 Click Test to view popup window with search results using the filter [Note: be sure your browser has popups enabled for Campus]

DL3400-1, DL3400-2. DL3400-3

Try It: Query Filter Search Options

- For text values
  - Equal or Not Equal (=, <>)
  - IN, NOT IN
  - LIKE
  - SOUNDS LIKE
  - IS NULL, IS NOT NULL
- For numeric values
  - >, >=, <, <=
- For dates
  - IS TODAY, IS YESTERDAY

Tip: The LIKE Option Uses Wildcards

% **matches a string of zero or more characters**

student.lastName	LIKE	Mc%	will find all names that begins with “Mc”
student.lastName	LIKE	%en%	will find all names that includes the letters “en” (Bennet, Green, McBadden)

_ **matches one character**

student.lastName	LIKE	Anders_n	will find “Anderson” and “Andersen” but not “Andersohn”

[ ] **matches each single character enclosed in the brackets**

student.lastName	LIKE	[ABCD]%	will find every name beginning with A, B, C or D
student.lastName	LIKE	[CKL]ars[eo]n	will find “Carsen”, “Karsen”, “Larsen”, “Carson”, “Karson” and “Larson”

^ **when used with square brackets, it means NOT**

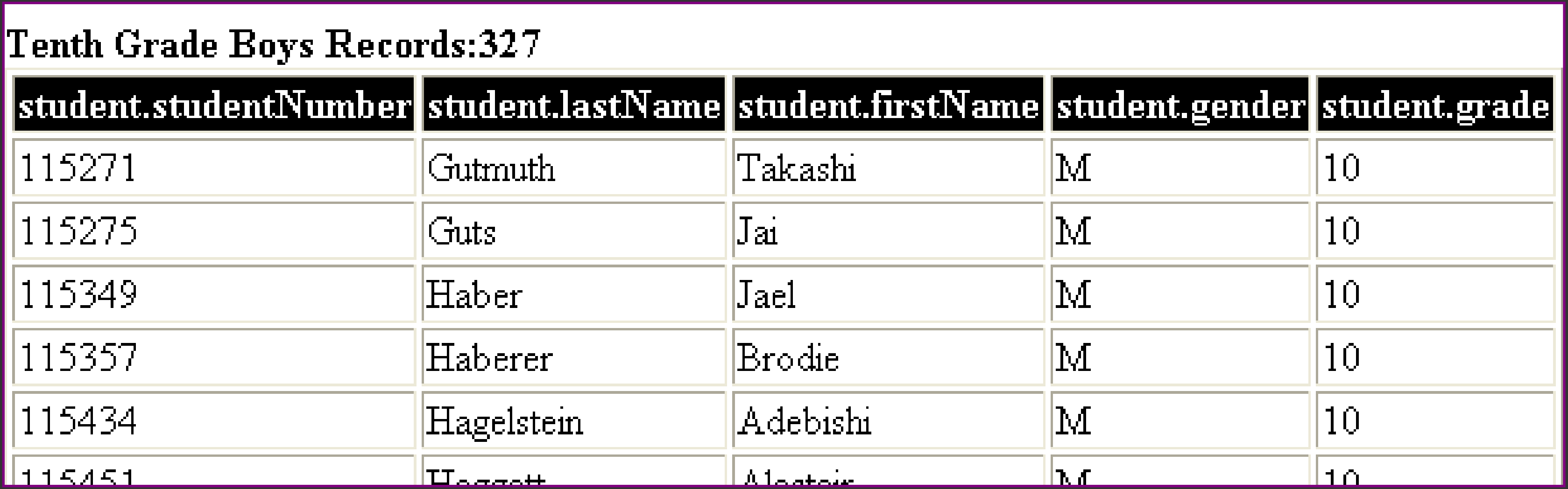
student.lastName	LIKE	M[^c]%	will find all names beginning with M that do not have c as the second letter

Try It: Create a Query Wizard Filter

- Create a Query Wizard filter
  - Filter name = SDQWXX##
    - XX = your initials
    - ## = last two digits of your phone number
  - Save it to your user account
- Examples
  - Filter students in grade 9 or 12
    - IN – using commas (no space)
    - student.grade IN 09,12
  - Filter on students with a last name that sounds like - as.
    - (Filter could return az, as, aas)
    - student.lastName SOUNDS LIKE as
  - Find students without certain information (i.e. birthdates, race ethnicity, middle name)
    - student.birthdate IS NULL
  - Find people who have unwanted information
    - Is Not Null
    - student.ssn IS NOT NULL

Try It: Query Wizard Check Your Work

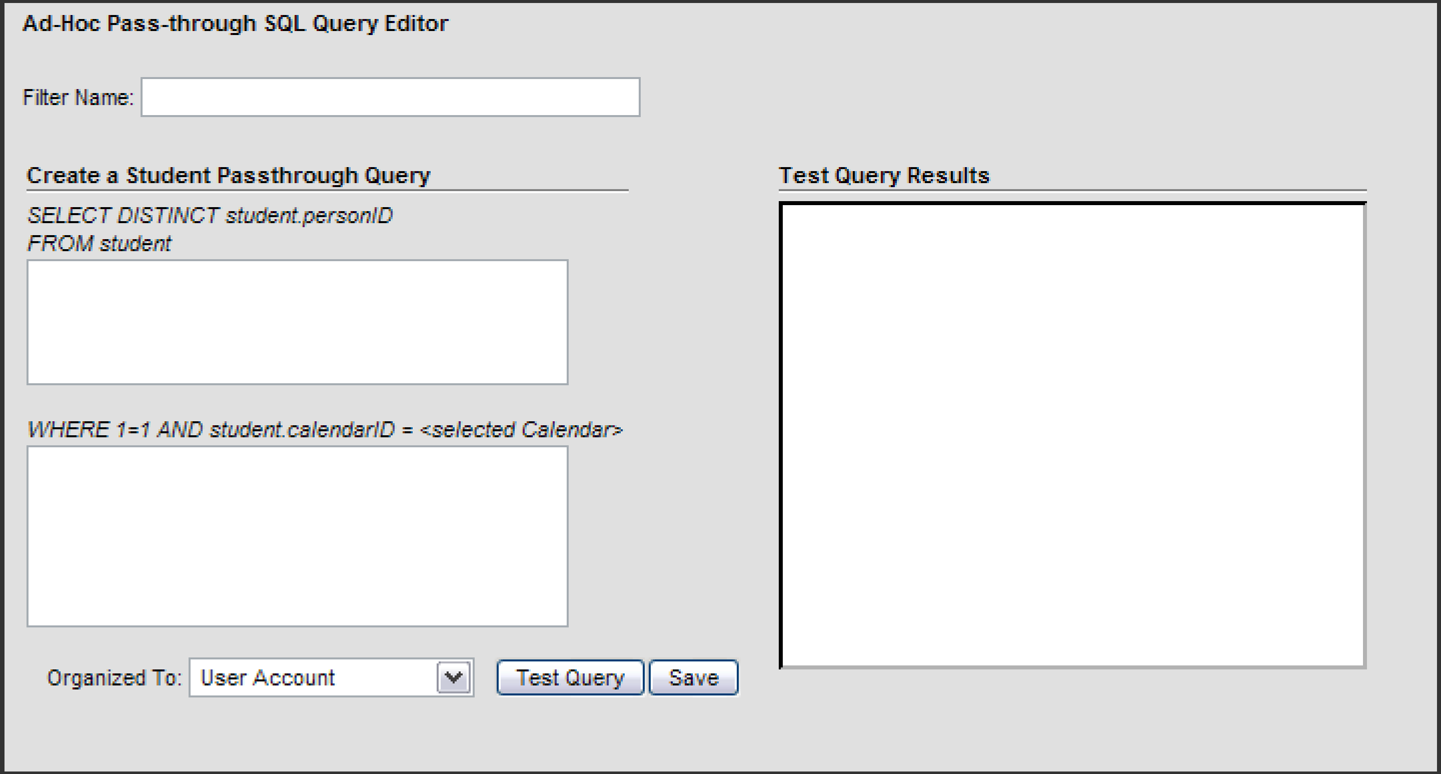
- Click Test to see results in a popup window
  - Make sure popups are allowed in your browser
  - Output can be copied and pasted into a spreadsheet if desired



Tenth Grade Boys Records:327

student.studentNumber	student.lastName	student.firstName	student.gender	student.grade
115271	Gutmuth	Takashi	M	10
115275	Guts	Jai	M	10
115349	Haber	Jael	M	10
115357	Haberer	Brodie	M	10
115434	Hagelstein	Adebishi	M	10
115451	Haggett	Alastair	M	10

SQL Pass-Through Queries



Ad-Hoc Pass-through SQL Query Editor

Filter Name:

Create a Student Passthrough Query

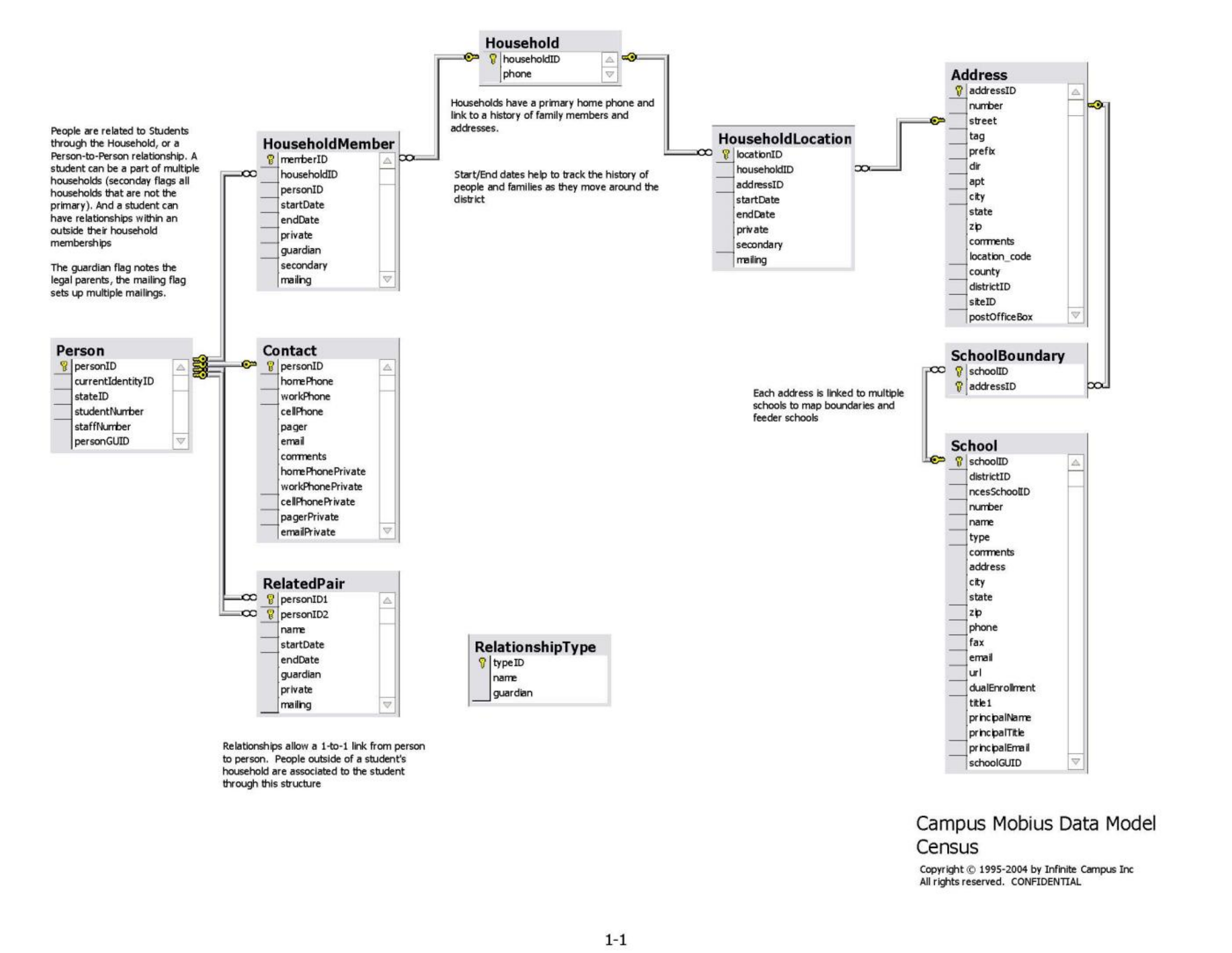
SELECT DISTINCT student.personID
FROM student

WHERE 1=1 AND student.calendarID = <selected Calendar>

Organized To: User Account

DL3400-1, DL3400-2. DL3400-3

Background - Campus Schema



People are related to Students through the Household, or a Person-to-Person relationship. A student can be a part of multiple households (seconday flags all households that are not the primary). And a student can have relationships within an outside their household memberships

The guardian flag notes the legal parents, the mailing flag sets up multiple mailings.

Household
- householdID
- phone

Households have a primary home phone and link to a history of family members and addresses.

HouseholdMember
- memberID
- householdID
- personID
- startDate
- endDate
- private
- guardian
- secondary
- mailing

Start/End dates help to track the history of people and families as they move around the district

HouseholdLocation
- locationID
- householdID
- addressID
- startDate
- endDate
- private
- secondary
- mailing

Address
- addressID
- number
- street
- tag
- prefix
- dir
- apt
- city
- state
- zip
- comments
- location_code
- county
- districtID
- siteID
- postOfficeBox

Person
- personID
- currentIdentityID
- stateID
- studentNumber
- staffNumber
- personGUID

Contact
- personID
- homePhone
- workPhone
- cellPhone
- pager
- email
- comments
- homePhonePrivate
- workPhonePrivate
- cellPhonePrivate
- pagerPrivate
- emailPrivate

SchoolBoundary
- schoolID
- addressID

Each address is linked to multiple schools to map boundaries and feeder schools

School
- schoolID
- districtID
- ncesSchoolID
- number
- name
- type
- comments
- address
- city
- state
- zip
- phone
- fax
- email
- url
- dualEnrollment
- title1
- principalName
- principalTitle
- principalEmail
- schoolGUID

RelatedPair
- personID1
- personID2
- name
- startDate
- endDate
- guardian
- private
- mailing

RelationshipType
- typeID
- name
- guardian

Relationships allow a 1-to-1 link from person to person. People outside of a student's household are associated to the student through this structure

Campus Mobius Data Model
Census

DL3400-1, DL3400-2. DL3400-3

Background – Relational Databases

- Relational Database
  - Multiple tables with records tied together by key fields
- Commonly used Campus key fields
  - calendarID
  - personID
  - courseID
  - sectionID

Understanding SQL

- The Campus engine runs on Microsoft SQL
- SQL reference books
- SQL cheat sheets/reference sites
  - www.cryer.co.uk/brian/sql/sql_crib_sheet.htm

The Grammar of SQL

- SQL statements have a very specific grammar

Format:

Select *fields*
From *table*
Where *table.field* [OPERANDI] [value]

Example:

Select *
From enrollment
Where enrollment grade='09'

DL3400-1, DL3400-2. DL3400-3

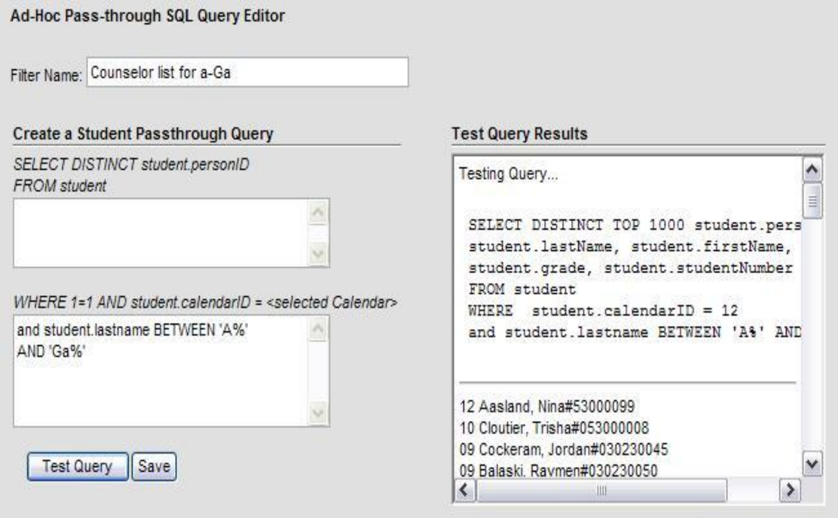
Pass-Through Parameters

IS NULL

IS NOT NULL

=, <>, >=, <=

BETWEEN



Ad-Hoc Pass-through SQL Query Editor

Filter Name: Counselor list for a-Ga

Create a Student Passthrough Query

SELECT DISTINCT student.personID
FROM student

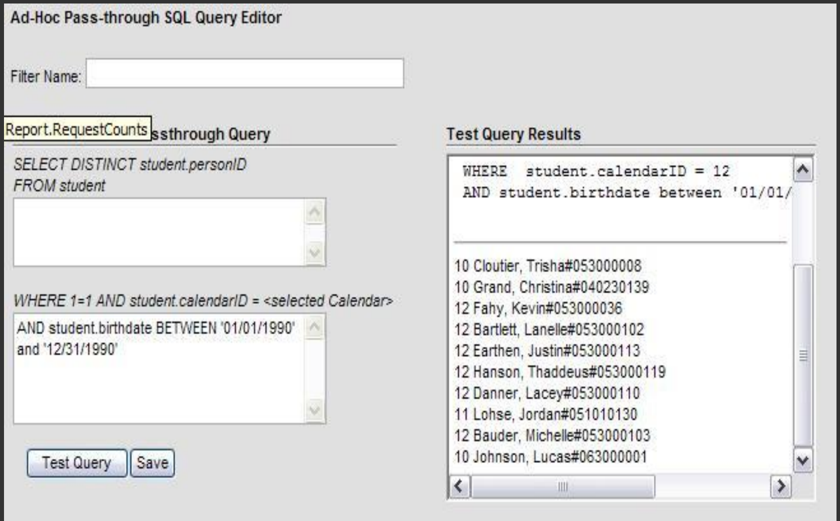
WHERE 1=1 AND student.calendarID = <selected Calendar>

and student.lastname BETWEEN 'A%'
AND 'Ga%'

Test Query | Save

Test Query Results

```
Testing Query...

 SELECT DISTINCT TOP 1000 student.pers
 student.lastName, student.firstName,
 student.grade, student.studentNumber
 FROM student
 WHERE  student.calendarID = 12
 and student.lastname BETWEEN 'A%' AND

12 Aasland, Nina#53000099
10 Cloutier, Trisha#053000008
09 Cockeram, Jordan#030230045
09 Balaski. Ravmen#030230050
```



Ad-Hoc Pass-through SQL Query Editor

Filter Name:

Report.RequestCounts ssthrough Query

SELECT DISTINCT student.personID
FROM student

WHERE 1=1 AND student.calendarID = <selected Calendar>

AND student.birthdate BETWEEN '01/01/1990'
and '12/31/1990'

Test Query | Save

Test Query Results

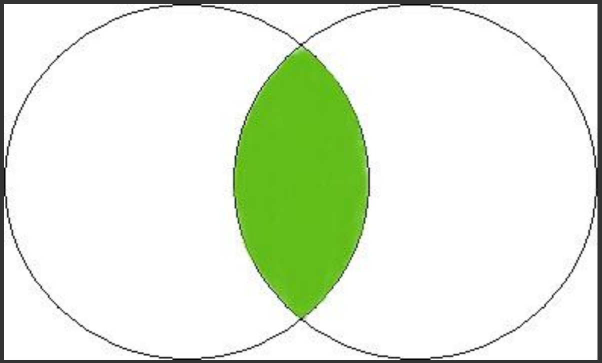
```
 WHERE  student.calendarID = 12
 AND student.birthdate between '01/01/

10 Cloutier, Trisha#053000008
10 Grand, Christina#040230139
12 Fahy, Kevin#053000036
12 Bartlett, Lanelle#053000102
12 Earthen, Justin#053000113
12 Hanson, Thaddeus#053000119
12 Danner, Lacey#053000110
11 Lohse, Jordan#051010130
12 Bauder, Michelle#053000103
10 Johnson, Lucas#063000001
```

DL3400-1, DL3400-2. DL3400-3

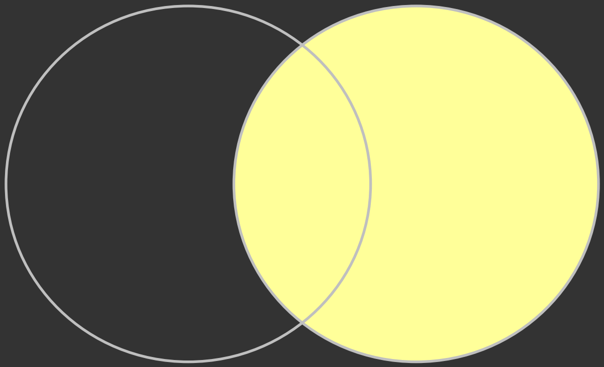
JOIN Statements

- SQL queries allow pulling data for the filter from a variety of tables by using a JOIN command
- INNER JOINS
  - Return all rows from multiple tables where the join condition is met
- OUTER JOINS
  - Returns all rows from one table and only those rows from a secondary table where the joined fields are equal (join condition is met)

Inner Join



Outer Join



INNER JOIN

- Allows tables with different data elements to be used in finding students
- Starting table is set based on data type
- Tables are joined based upon having a key match

Format:

```
From table1
            Inner join table2 on table1.field =table2.field
Where table2.field = X AND table1.field=Y
```

DL3400-1, DL3400-2. DL3400-3

Uses of INNER JOIN in SQL

- 8th graders going to a particular high school (multi-HS district) based on
  - Address (school boundary)
  - Next calendar, next grade (current enrollment)

OUTER JOIN

- Use OUTER JOIN to exclude records
- Example
  - To see all 9th grade students who ***do not*** have requests in the system
  - Use a left outer join to exclude students who ***do*** have requests

Format:

```
Select table1.[field], table1.[field]
From table1
        Left outer join table2 on table1.field =table2.field
Where tableN.field = X
```

Example:

```
From student
        left outer join request r on s.personid = r.personid
where r.requestid is null
```

DL3400-1, DL3400-2. DL3400-3

Examples of SQL Queries

- Students in 12th Grade with less than 20 credits

```
A: INNER JOIN v_transcriptdetail v on v.personID = student.personID
B: AND student.grade = '12'
        GROUP BY student.personID,
        student.lastName, student.firstName,
        student.grade, student.studentNumber
        HAVING(COUNT(v.creditsearned)) <20
```

- Birthdates

```
A: Nothing
B: AND student.birthDate BETWEEN '5/1/1991' AND '5/31/2000‘
```

Examples of SQL Queries

- Courses w/out a certain task

```
A: INNER JOIN gradingtaskcredit gtc on gtc.courseID =
course.courseID AND gtc.calendarID = course.calendarID
B: AND gtc.taskID <> 56
```

- Requested Course that students did not get

```
A:  INNER JOIN request r on r.personID = student.personID AND
r.calendarID = student.calendarID
INNER JOIN course c on c.courseID = r.courseID and c.number
='0355'
INNER JOIN [section] s on s.courseID = c.courseID
LEFT OUTER JOIN roster ro on ro.personID = r.personID AND
ro.sectionID =
s.sectionID
B: AND ro.personID IS NULL
```

Questions & Answers

Ask, we’re ready!

Learn More!

Additional training is available from Campus U

- Professional, certified trainers
- Just-in-time offerings
- Online
- In person
  - In your district
  - At Infinite Campus

Infinite Campus University